



HELLO  
WORLD



# COMPONENTES EM REACT

# CRIANDO UM PROJETO REACT DE FORMA AUTOMATIZADA

## Criando um Projeto React de Forma Automatizada:

A maneira mais eficiente de criar um projeto React atualmente é utilizando o comando create-react-app ou ferramentas mais modernas como Vite, que oferece maior desempenho e simplicidade. Aqui usaremos o Vite, uma alternativa leve e rápida.

01

### Instale o Vite utilizando o comando:

- `npm create vite@latest my-react-app --template react`

02

### Acesse o diretório do projeto:

- `cd my-react-app`

03

### Instale as dependências:

- `npm install`

04

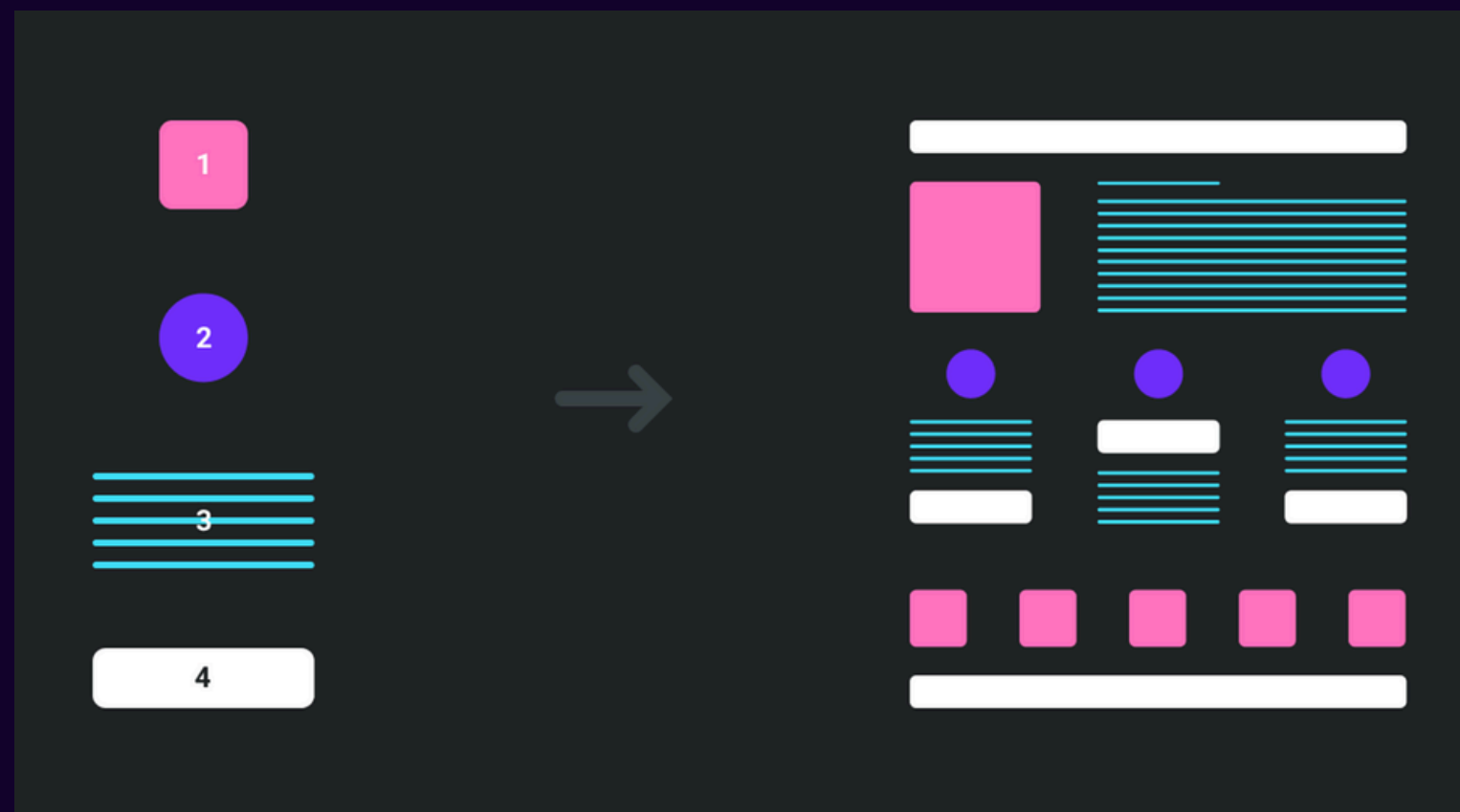
### Inicie o servidor de desenvolvimento:

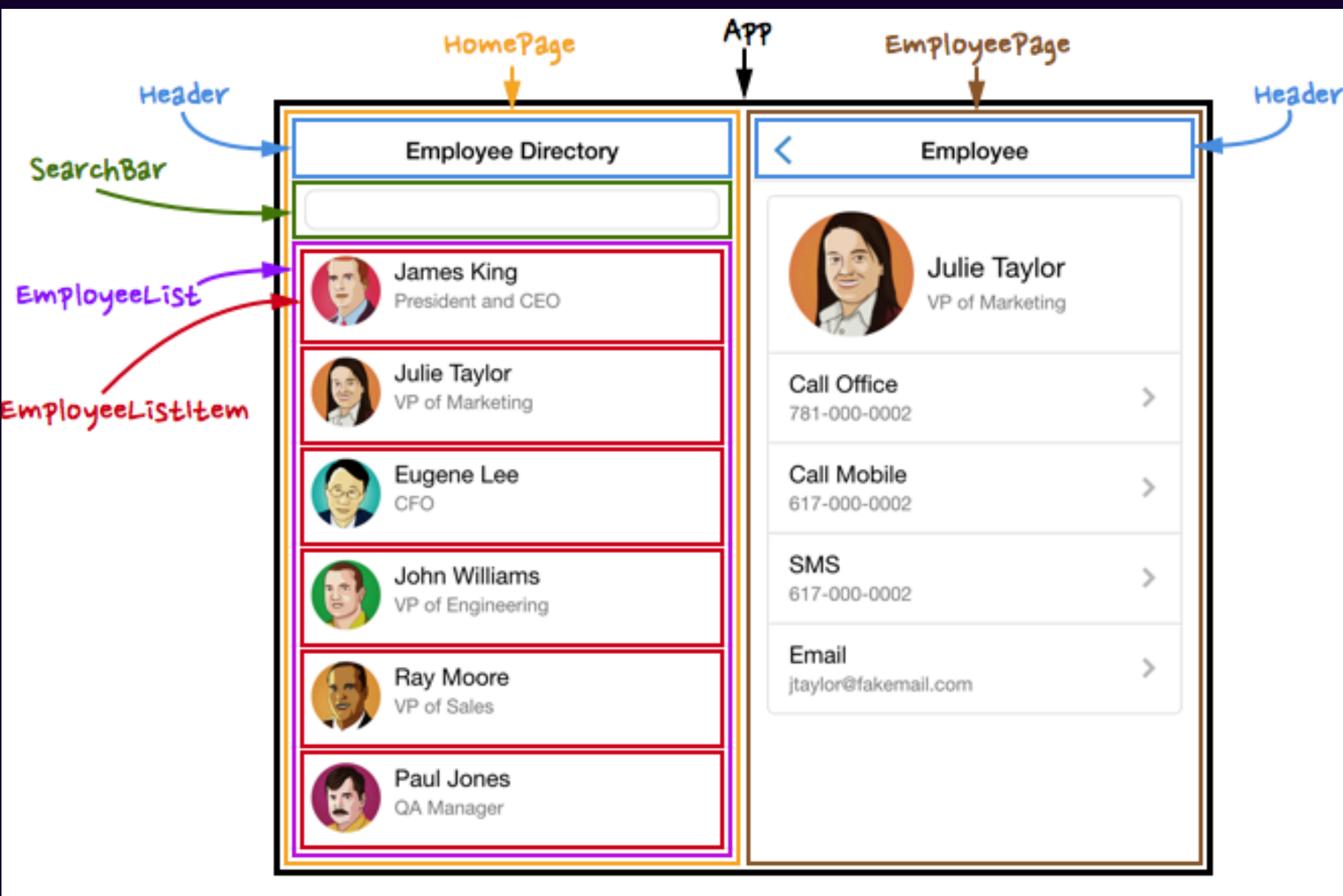
- `npm run dev`

# INTRODUZINDO O CONCEITO DE COMPONENTES

## O Que São Componentes?

Componentes são blocos reutilizáveis de código que representam partes da interface do usuário. Eles permitem que você divida sua aplicação em pequenas partes independentes e reutilizáveis, facilitando a manutenção e escalabilidade do projeto.





# COMPONENTES

Os componentes são as unidades fundamentais do React, representando partes isoladas da interface.

01

**Reutilizáveis**

02

**Isolados:** Alterar um componente não afeta os outros

03

**Combináveis:** Vários componentes podem ser usados juntos para criar interfaces complexas



# TIPOS DE COMPONENTES

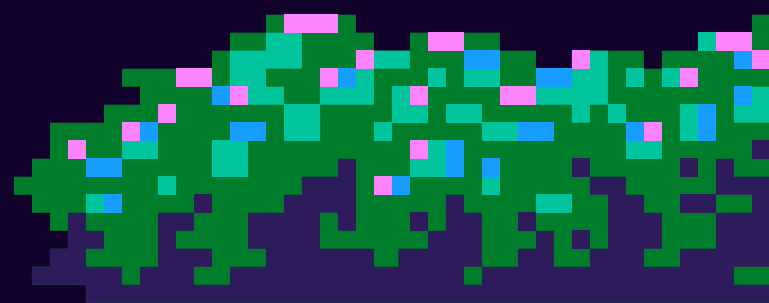
**Componentes funcionais:** Criados como funções JavaScript que retornam elementos HTML.

```
1 function Botao() {  
2   return <button>Click me!</button>;  
3 }
```

VS

**Componentes de classe:** Herdam de `React.Component` e utilizam o método `render()`.

```
1 class Botao extends React.Component {  
2   render() {  
3     return <button>Click me!</button>;  
4   }  
5 }
```





# ESTILOS INLINE EM JSX

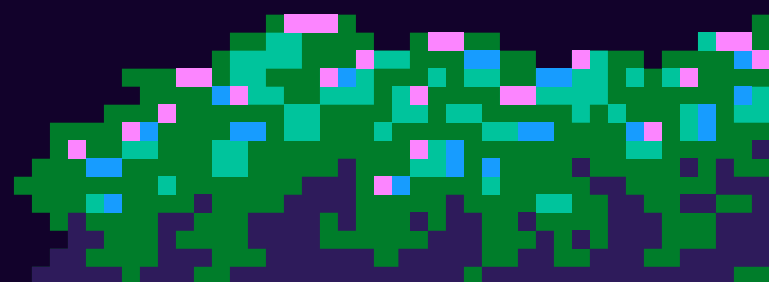


No React, você pode aplicar estilos diretamente aos elementos HTML usando o atributo `style`. O valor de `style` deve ser um objeto JavaScript onde os nomes das propriedades CSS são escritos em camelCase.



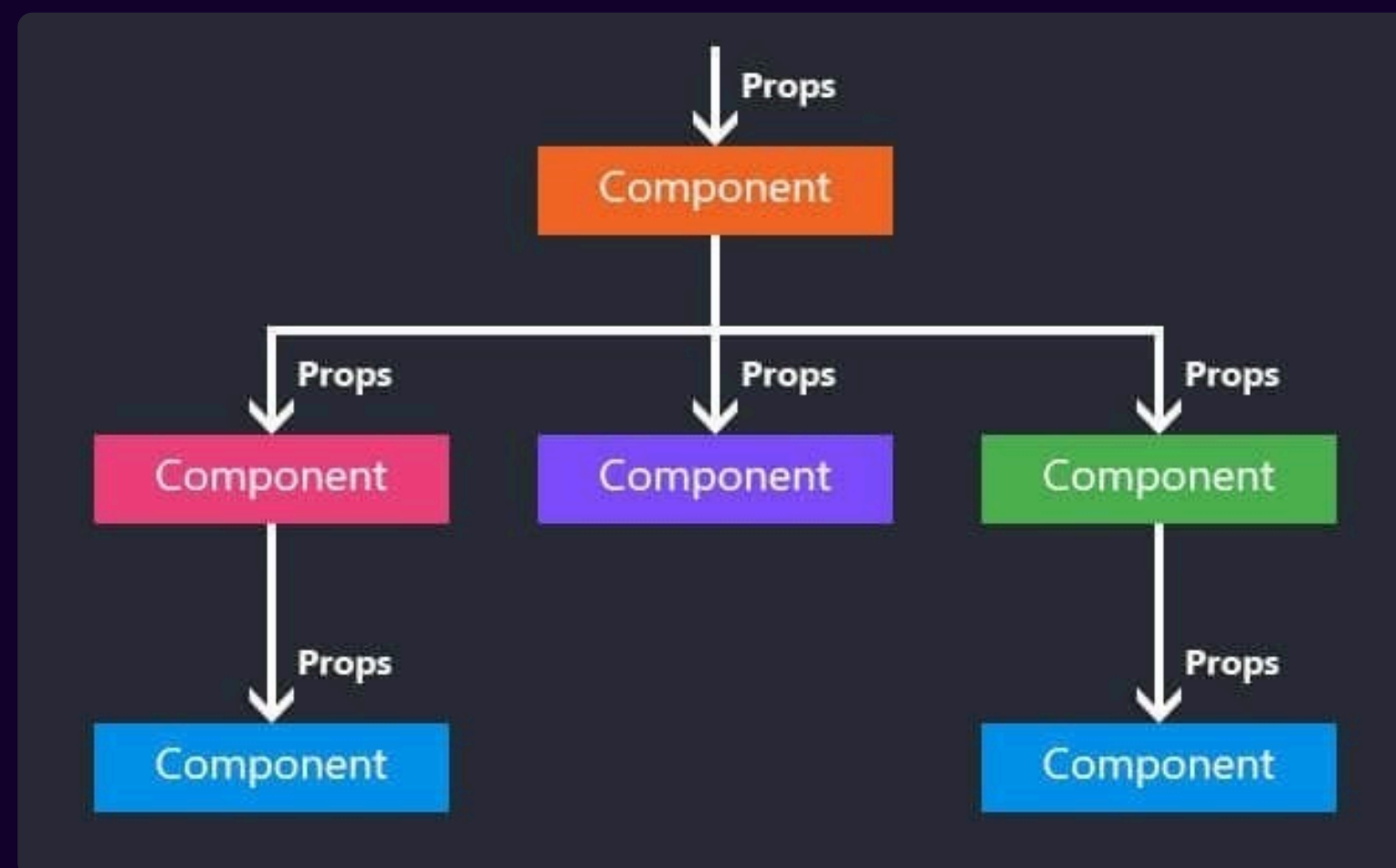
```
1 function Header({ title }) {  
2   return (  
3     <header style={{ backgroundColor: '#282c34', color: 'white', padding: '20px', textAlign: 'center' }}>  
4       <h1>{title}</h1>  
5     </header>  
6   );  
7 }  
8  
9 export default Header;
```

## Observações:

- 1.As propriedades CSS compostas, como **background-color**, devem ser escritas como **backgroundColor**.
  - 2.Os valores numéricos, como **margin** ou **padding**, podem ser passados diretamente sem aspas. Valores como cores precisam ser strings.
  - 3.Embora útil para estilos rápidos, é recomendável utilizar outras abordagens, como classes **CSS**, para manter a organização do código.
- 

# O QUE SÃO PROPRIEDADES?

Os componentes podem receber dados dinâmicos através de "props" (abreviação de properties). As props são passadas para os componentes como atributos.







EODRA

AEAGIA

# NOSSOS CONTATOS:



27 99500-7495



<https://beacons.ai/prismatech>



[producaoprismatech@gmail.com](mailto:producaoprismatech@gmail.com)



Avenida Jerônimo Monteiro 145, Vitória





THANK  
YOU